كلية المأمون الجامعة

Computer Science Department

Evolutionary Computing & Machine Learning

د طيف سامي حسن

2017-2018

Introduction to Machine Learning

- It is algorithm used for learning to do stuff
- is a field of computer science that uses statistical techniques to give computer systems the ability to "learn with data, without being explicitly programmed

Example of Machine Learning

- Optical character recognition
- Face detection
- Spam filtering
- Topic spotting
- Spoken language understanding
- Medical diagnosis
- Customer segmentation
- Fraud detection
- Weather prediction

Goal of Machine Learning

Primary goal→ develop general purpose algorithm of practical value

Advantage of machine learning over direct programming

- ML Result more accurate
- ML: human expert guided by imprecise impression
- ML : data driven

Machine Learning model

- It is formally define learning problem
- Model must be rich enough to capture important aspect of real learning problem
- Model must be simple enough to study problem mathematically

Concept learning Task

- Most general hypothesis (?)
- Most specific hypothesis (ϕ)

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	War,	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	sunny	Warm	High	strong	cool	change	yes

Concept learning as search

CL→ search through a large space of hypothesis implicitly

***FIND-S*: FINDING THE MOST SPECIFIC HYPOTHESIS**

1. Initialize h to the most specific hypothesis in H
2. For each positive training instance x
   - For each attribute constraint a_i in h
     - If the constraint a_i is satisfied by x
     - Then do nothing
     - Else replace a_i in h by the next more general constraint that is satisfied by x
3. Output hypothesis h

FIND-S Algorithm.

Step 1: FIND-S

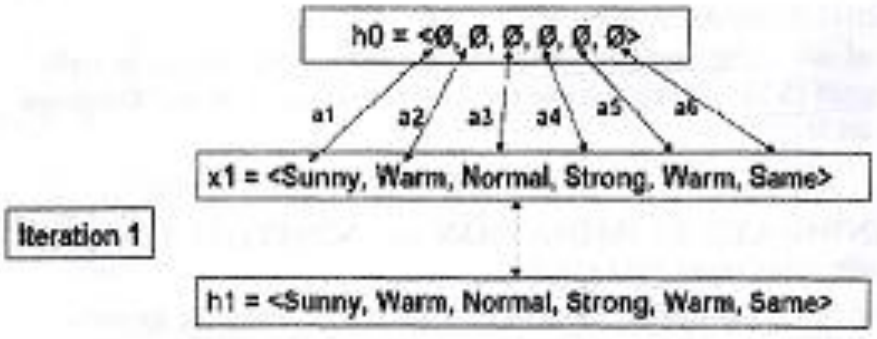
Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

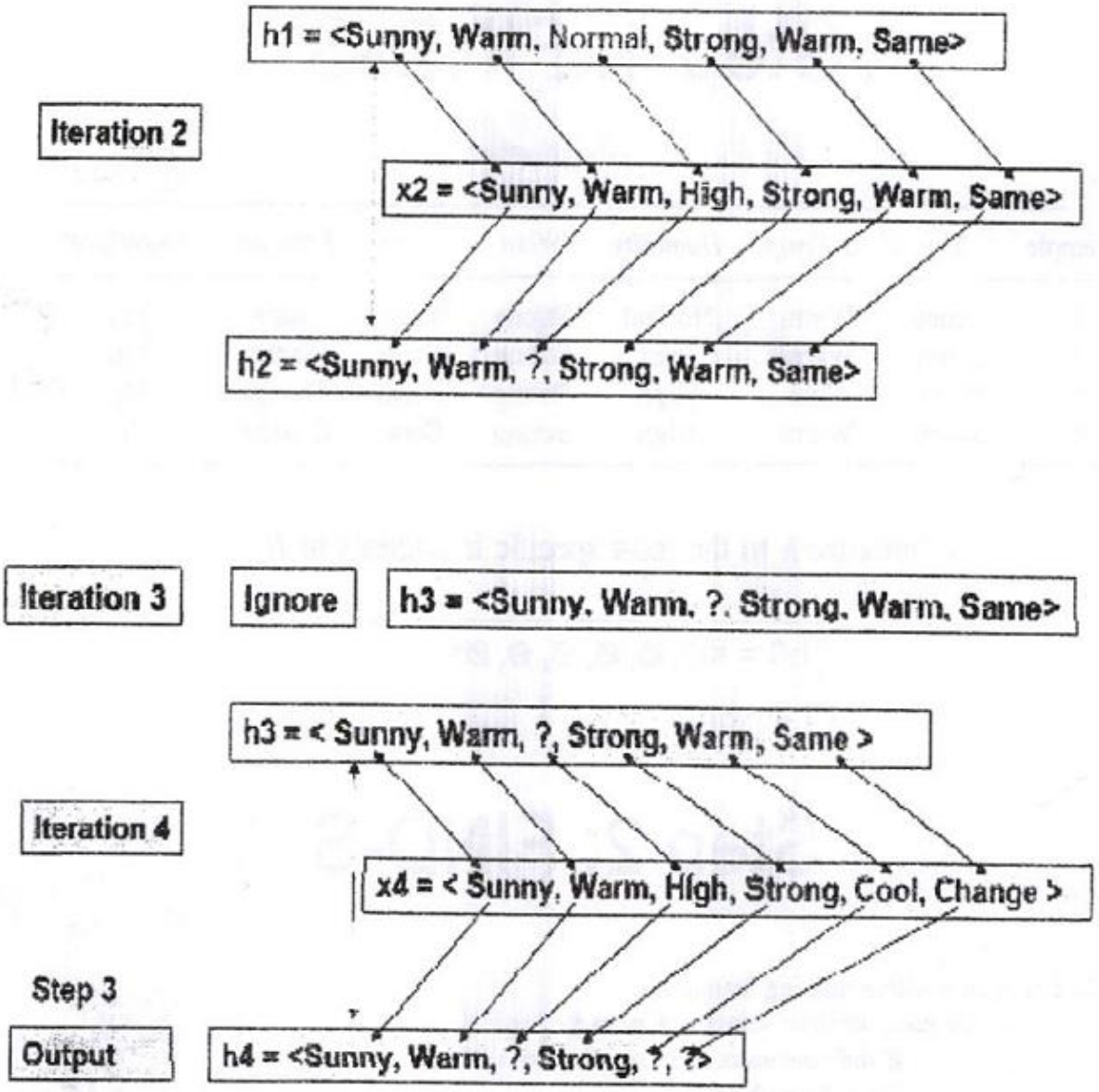
1. Initialize h to the most specific hypothesis in H

h0 = <Ø, Ø, Ø, Ø, Ø, Ø>

Step 2: FIND-S

2. For each positive training instance x
   - For each attribute constraint a_i in h
     If the constraint a_i is satisfied by x
     Then do nothing
     Else replace a_i in h by the next more general constraint that is satisfied by x





Version Space

Set of all valid hypothesis provide by an algorithm

The LIST-THEN-ELIMINATE Algorithm

1. *VersionSpace* ← a list containing every hypothesis in H
2. For each training example, $\langle x, c(x) \rangle$

   remove from *VersionSpace* any hypothesis h for which $h(x) \neq c(x)$
3. Output the list of hypotheses in *VersionSpace*

Candidate-Elimination Algorithm

Initialize G to the set of maximally general hypotheses in H
Initialize S to the set of maximally specific hypotheses in H
For each training example d, do

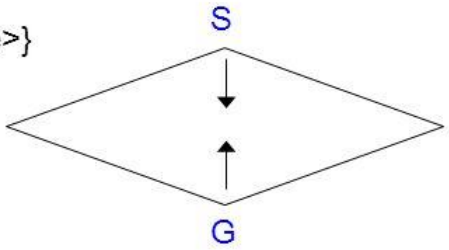
- If d is a positive example
  - Remove from G any hypothesis inconsistent with d
  - For each hypothesis s in S that is not consistent with d
    - Remove s from S
    - Add to S all minimal generalizations h of s such that
      - h is consistent with d, and some member of G is more general than h
    - Remove from S any hypothesis that is more general than another hypothesis in S
- If d is a negative example
  - Remove from S any hypothesis inconsistent with d
  - For each hypothesis g in G that is not consistent with d
    - Remove g from G
    - Add to G all minimal specializations h of g such that
      - h is consistent with d, and some member of S is more specific than h
    - Remove from G any hypothesis that is less general than another hypothesis in G

Candidate-Elimination Algorithm

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

S_0 = {<∅, ∅, ∅, ∅, ∅, ∅>}
G_0 = {<?, ?, ?, ?, ?, ?>}

S_1 = {<Sunny, Warm, Normal, Strong, Warm, Same>}
G_1 = {<?, ?, ?, ?, ?, ?>}



S_2 = {<Sunny, Warm, ?, Strong, Warm, Same>}
G_2 = {<?, ?, ?, ?, ?, ?>}

S_3 = {<Sunny, Warm, ?, Strong, Warm, Same>}
G_3 = {<Sunny, ?, ?, ?, ?, ?>, <?, Warm, ?, ?, ?, ?>, <?, ?, ?, ?, ?, Same>}

S4 = {<Sunny, Warm, ?, Strong, ?, ?>}
G4 = {<Sunny, ?, ?, ?, ?, ?>, <?, Warm, ?, ?, ?, ?>}

41

Candidate Elimination: example

S: {<∅, ∅, ∅, ∅, ∅, ∅ >}

G: {<?, ?, ?, ?, ?, ?>}

x_1 = <Sunny Warm Normal Strong Warm Same> +

S: {< Sunny Warm Normal Strong Warm Same >}

G: {<?, ?, ?, ?, ?, ?>}

x_2 = <Sunny Warm High Strong Warm Same> +

S: {< Sunny Warm ? Strong Warm Same >}

G: {<?, ?, ?, ?, ?, ?>}

Candidate Elimination: example

S: {< Sunny Warm ? Strong Warm Same >}

G: {<?, ?,?, ?, ?, ?>}

x_3 = <Rainy Cold High Strong Warm Change> -

S: {< Sunny Warm ? Strong Warm Same >}

G: {<Sunny,?,?,?,?,?>, <?,Warm,?,?,?>, <?,?,?,?,?,Same >

x_4 = <Sunny Warm High Strong Cool Change> +

S: {< Sunny Warm ? Strong ? ? >}

G: {<Sunny,?,?,?,?,?>, <?,Warm,?,?,?>}

ايجاد جميع العناصر الصغيرة محذوفا منها العناصر التي يتم فيها التطابق (S مع X) وعند عدم المطابقة نختار S

مثلا جميع العناصر هي

<Sunny,?,?,?,?,?>, <Rainy,?,?,?,?,?> ✗

<?,Warm,?,?,?,?>, <?,Cold,?,?,?,?> ✗

<?,?,strong,?,?,?> ✗

✗ :?,?,?,Warm,?,?>, <?,?,?,Cool,?,?> ✗

<?,?,?,?,?,Same>, <?,?,?,?,?,Change> ✗

يقارن G3 مع X4 ويحذف من G3 العناصر التي لاتتطابق والناتج هو G4